Pixel Motion Diffusion is What We Need for Robot Control

Supplementary Material

1. Training Details

We train all models on a single node with 4 NVIDIA A6000 GPUs. For Motion Director, we train for 100k iterations with a per-GPU batch size of 16. For Action Expert, we train for 10k iterations with a per-GPU batch size of 64. We use the AdamW optimizer with a learning rate of 1×10^{-4}. Mixed precision training is used to reduce memory usage and improve throughput. All training is implemented in PyTorch with the HuggingFace Diffusers and Transformers libraries.

2. Dataset details

2.1. CALVIN

Dataset: CALVIN is an open source simulated benchmark to learn long-horizon language-conditioned tasks, which contains 4 different simulation environments-A, B, C, D. While each split (A–D) shares the same robotic setup, variations in object placement, textures, lighting, and distractors ensure that models cannot rely on memorization but must instead demonstrate robust visuomotor understanding. The 34 manipulation tasks span a wide range of skills such as pushing, placing, rotating, toggling switches, and opening drawers, all expressed through natural language instructions.

Evaluation: We follow standard evaluation protocol from [1], which evaluates a given policy on 1000 episodes each containing 5 continuous tasks (i.e. task i starts from the end state of task $i-1$, which is often different to what is encountered in demonstrations within the training data). For each task, at most 360 action steps are performed unless the task is successfully completed prior to that. The success rate for each consecutive task is averaged across the 1000 episodes and reported. Considering the 5 continuous tasks as a sequences, the average number of tasks completed by the policy (i.e. average length) is also reported.

2.2. DROID

DROID is a large-scale "in-the-wild" robot manipulation dataset featuring 76k real demonstration trajectories across 564 varied scenes and 86 tasks. It provides over 350 hours of interaction data, with diverse viewpoints, object types, and natural instruction annotations.

2.3. Real World

2.3.1. xArm Environment

We constructed a dataset specifically for fine-tuning and real-world evaluation. The experimental platform consists of a 7-DoF xArm7 manipulator and two RGB cameras. An Intel RealSense D435 was positioned laterally to provide a third-person view of the workspace, while an Intel RealSense D405 was mounted above the gripper to capture a close-up view of the end-effector and its interactions with objects. Though both cameras are stereo cameras, we only use a single RGB view from each camera in all the experiments. This dual-camera setup enables complementary perspectives, facilitating both scene-level and fine-grained observations.

Data was collected through a leader–follower teleoperation scheme, where a human operator controlled a leader device to guide the motions of the xArm7 (follower). Each demonstration episode was restricted to a single atomic task, such as lifting a fruit, transporting it, or placing it into a basket. Episodes were initialized either from randomized joint configurations or from the terminal state of the preceding task, ensuring diversity in initial conditions. To further increase variability and promote generalization, we occasionally re-dropped and re-grasped objects within the same episode.

The resulting dataset comprises 1,000 episodes, with a minimum of 100 demonstrations allocated to each distinct task. This distribution ensures both task balance and sufficient coverage for downstream fine-tuning. Overall, the dataset provides a structured yet diverse collection of manipulation trajectories suitable for evaluating task-specific policies under realistic conditions.

2.3.2. Galaxea R1 Lite Environment

In order to evaluate on more complicated environment, we also constructed a dataset for fine-tuning and real-world evaluation. This environment contains a bi-manual Galaxea R1 Lite platform and three RGB cameras. A RGB camera was positioned medially to provide the head view of the workspace, while the other two Intel Realsense D405 was mounted above two grippers in each side to capture the close-up view of the end-effector and their interactions with objects. Though both of the gripper-view cameras are stereo cameras, we only use a single RGB view from each camera in all the experiments.

We set up a table top manipulation environment, which has two plush toys: a lion and an elephant, also a toy pot. In terms of the tasks, we are manipulating plush toys and the lid cover of the toy pot (one hand manipulates a plush toy, and the other hand helps open the lid cover of the toy pot). See Figure 3 for the example and Table 1 for the task description.

We collected our data through a leader-follower teleoperation scheme, which human operates the Galaxea R1 Lite Teleop to guide the motions of the Galaxea R1 Lite (follower). Each demonstration episode was a single bi-manual

Task	Language
left_place_lion	left arm lift and place lion into pot, right arm helps open and close lid.
left_place_elephant	left arm lift and place elephant into pot, right arm helps open and close lid.
left_pick_lion	left arm pick lion from pot, right arm helps open and close lid.
left_pick_elephant	left arm pick elephant from pot, right arm helps open and close lid.
right_place_lion	right arm lift and place lion into pot, left arm helps open and close lid.
right_place_elephant	right arm lift and place elephant into pot, left arm helps open and close lid.
right_pick_lion	right arm pick lion from pot, left arm helps open and close lid.
right_pick_elephant	right arm pick elephant from pot, left arm helps open and close lid.

Table 1. Task definitions used in our Galaxea Real World experiments.

pick-place task, such as picking the lion into the pot and helping to open the lid. The collected dataset comprises 150 episodes, which provides a structured collection of learnable bi-manual manipulation trajectories.

3. More analysis

3.1. Computational trade-off

As shown in Table 2, reducing the number of reverse diffusion steps in the Motion Director during inference substantially lowers the latency, with performance degrading slightly. This demonstrates a practical and potential mechanism to support higher-frequency control by trading off motion fidelity for speed.

Diffusion Steps	Latency(ms)	Avg. Len
2	32	3.88
10	60	3.96
25	99	**4.00**
40	149	3.95

Table 2. Trade-off between reverse diffusion steps on Motion Director and performance on CALVIN validation dataset.

3.2. Motion Director metrics

We include a quantitative comparison of pixel motion prediction error (MSE) against ground-truth optical flow between DAWN and LTM (Table 3). DAWN achieves lower prediction error, indicating much more accurate pixel motion quality. In addition, Table 6 (a) in the paper shows that using pixel motion consistently outperforms RGB or no-motion variants.

Model	Error
LTM	5.19×10^{-4}
DAWN	1.23×10^{-4}

Table 3. Motion Director pixel motion prediction error (MSE) on CALVIN validation dataset.

3.3. Ablation study on temporal offset k

Table 4 shows that performance is stable across different k, with k=20 yielding the best results and we use it throughout all our experiments.

k	5	10	20	30
Avg. Len	3.72	3.93	**4.00**	3.67

Table 4. Ablation study of different temporal offset k in CALVIN environment.

4. Qualitative Results

This section includes a series of visualizations demonstrating how DAWN generates and executes pixel-motion plans across diverse environments. All videos and overlays are packaged locally and can be viewed through the provided index.html together with this supplementary.

4.1. Bi-manual Pixel Motion Predictions

We first showcase Motion Director's pixel-motion predictions on bimanual manipulation sequences. These include both our own recorded bimanual setup and Galaxea-Open-World-Dataset videos. In each case, we overlay the predicted pixel motion on each frame to reveal how the Motion Director captures coordinated left–right arm movements, object-relative displacements, and long-range motion cues. These examples highlight that the motion plans remain consistent even in visually complex or asymmetric dual-arm settings. Our two examples of Galaxea-Open-World-Dataset pixel motion prediction are presented in Figure 2. Another two examples of our real world environment pixel motion prediction are presented in Figure 3.

4.2. Real-World xArm Manipulation

Next, we provide full rollout videos from our real-world xArm7 platform (see Figure 1). For every rollout, we include third-person view and frame-by-frame pixel-motion overlays. These visualizations show that the robot's actual behavior reliably follows the predicted motion. This makes the high-level plan interpretable, which is one of the key advantages of using structured pixel-motion as the intermediate representation.

4.3. CALVIN Rollout

We also include additional CALVIN rollout examples with paired RGB frames and predicted pixel motions. Similar

to the real-world experiments, Motion Director produces clean, directional pixel-motion fields, and Action Expert executes them through temporally coherent low-level actions. These long-horizon sequences further confirm the consistency between planned and executed motion, even when the tasks involve multi-object interactions, distractors, or ambiguous scene layouts. Our two example rollouts are presented in Figure 4.

References

[1] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, and et al. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv:2412.14803*, 2024.

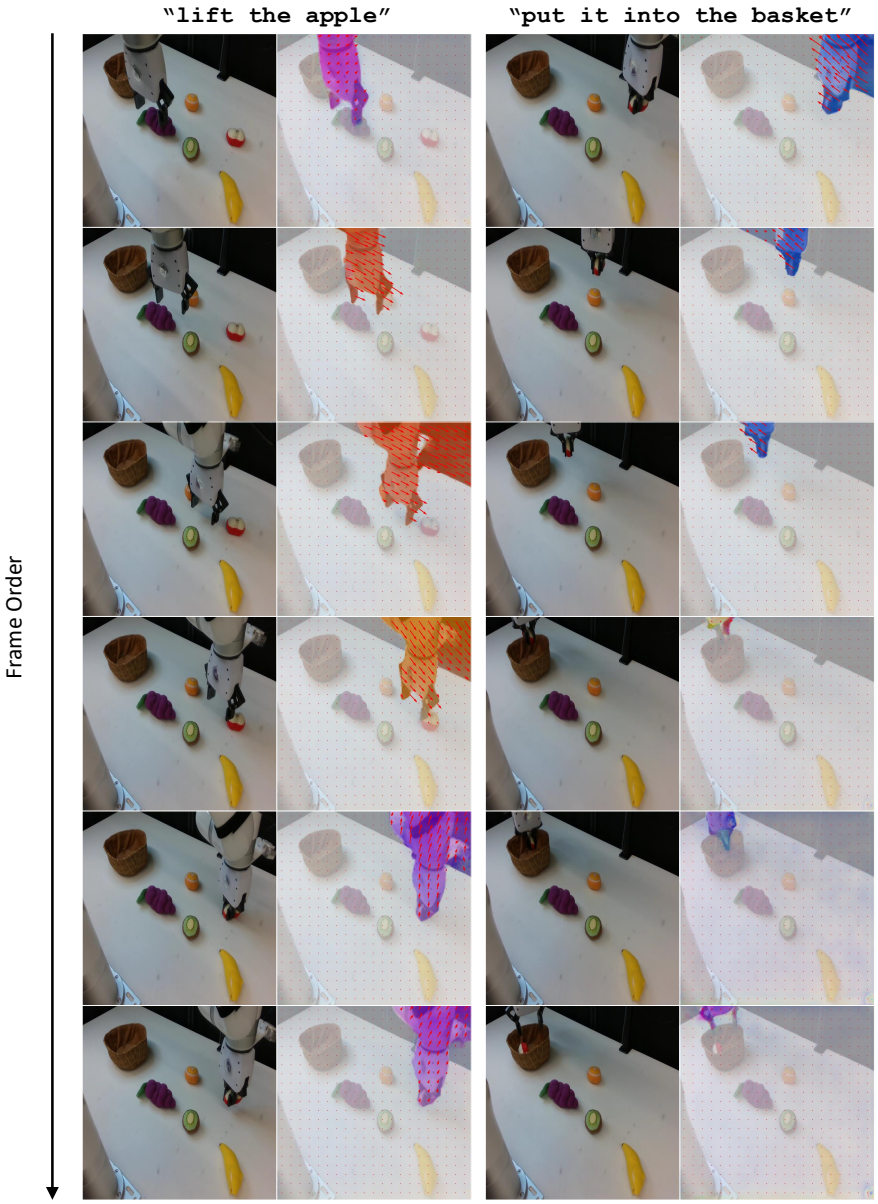

Figure 1. **xArm rollout examples**. The first column shows the observation sequence given the task of "lift the apple". The second column shows the observation sequence given the task of "put it into the basket". Each group shows the original static-camera observation and the visualizations of corresponding pixel motions predicted by Motion Director.

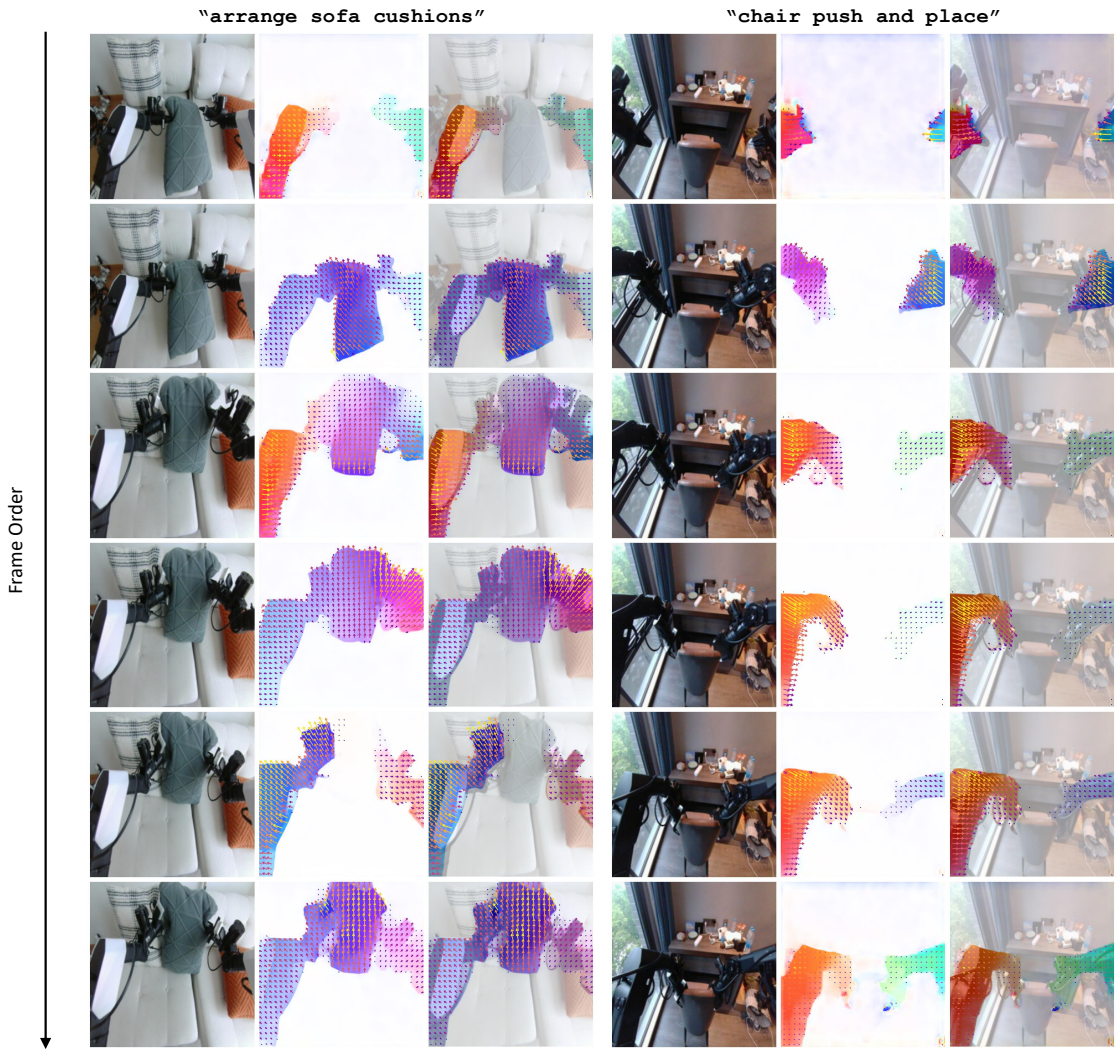

Figure 2. **Galaxea-Open-World-Dataset pixel motion prediction examples**. The first column shows the one test image sequence given the task of "arrange sofa cushions". The second column shows the test image sequence given the task of "chair push and place". Each group shows the original head-camera observation and the visualizations of corresponding pixel motions predicted by Motion Director.

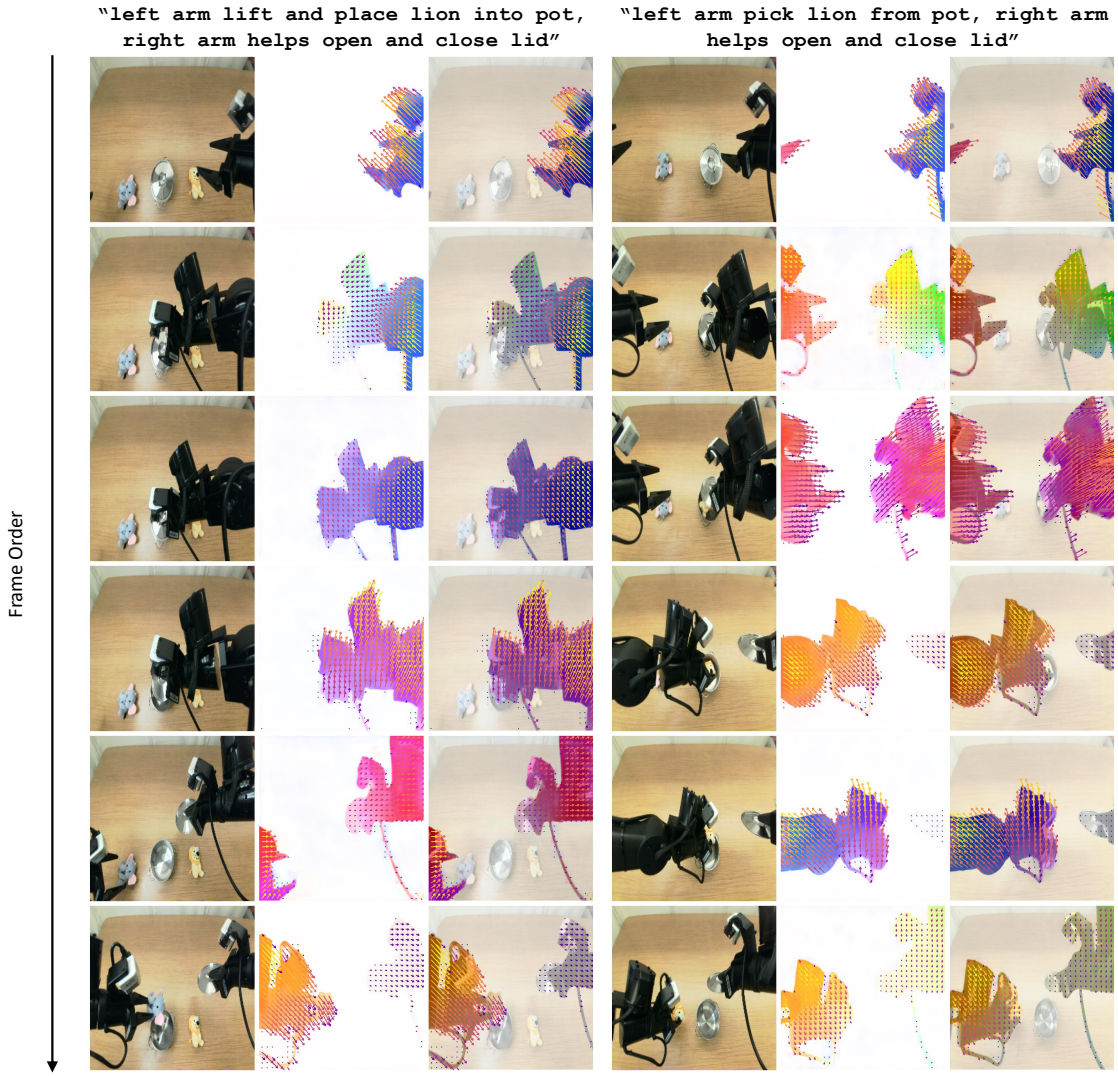

Figure 3. **Real-World Galaxea pixel motion prediction examples**. The first column shows the one test image sequence given the task of "left arm lift and place lion into pot, right arm helps open and close lid". The second column shows the test image sequence given the task of "left arm pick lion from pot, right arm helps open and close lid". Each group shows the original head-camera observation and the visualizations of corresponding pixel motions predicted by Motion Director.

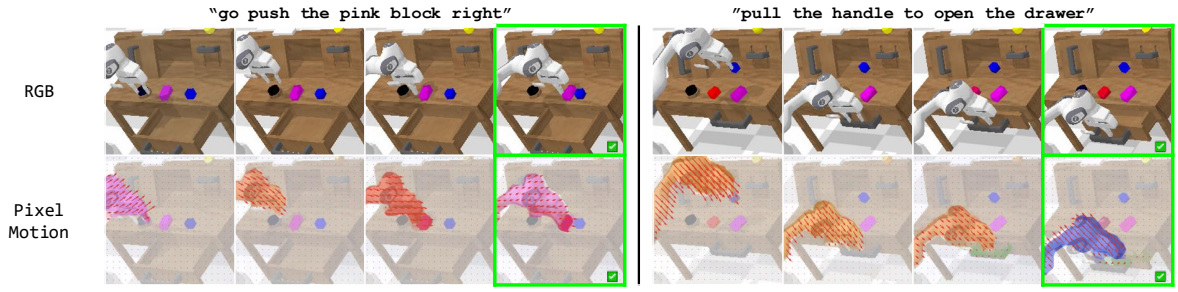

Figure 4. **CALVIN rollout examples**. Two example rollouts of DAWN in CALVIN environment. The first row is the sequence of RGB images, and the second row is the visualization of the corresponding pixel motions predicted by Motion Director.